

# The generalized finite point method

B. Boroomand · M. Najjar · E. Oñate

Received: 8 October 2008 / Accepted: 3 January 2009 / Published online: 3 February 2009  
© Springer-Verlag 2009

**Abstract** In this paper we propose a new mesh-less method based on a sub-domain collocation approach. By reducing the size of the sub-domains the method becomes similar to the well-known finite point method (FPM) and thus it can be regarded as the generalized form of finite point method (GFPM). However, unlike the FPM, the equilibrium equations are weakly satisfied on the sub-domains. It is shown that the accuracy of the results is dependent on the sizes of the sub-domains. To find an optimal size for a sub-domain we propose a patch test procedure in which a set of polynomials of higher order than those chosen for the approximations/interpolations are used as the exact solution and a suitable error norm is minimized through a size tuning procedure. In this paper we have employed the GFPM in elasto-static problems. We give the results of the size optimization in a series of tables for further use. Also the results of the integrations on a generic sub-domain are given as a series of library functions for those who want to use GFPM as a cheap and fast integral-based mesh-less method. The performance of GFPM has been demonstrated by solving several sample problems.

**Keywords** Mesh less · Finite point method · Sub-domain · Collocation · Patch test

## 1 Introduction

Along with the advances in mesh-based methods such as finite element method, mesh-less methods are also growing and developing fast. The aim is to make the numerical analyses as cheap as possible and this, however, has some drawbacks when compared with the well developed mesh-based approaches. The development history of the mesh-less methods can be found in many papers [1–4] and thus we avoid mentioning it here.

Similar to other numerical methods, construction of a mesh-less method requires making decision on two basic parts, i.e. the form of the approximation/interpolation and the formulation for approximately satisfying the governing equations. Several approximation/interpolation schemes are available in the literature; among them are the least squares (LS), weighted least squares (WLS), moving least squares (MLS), reproducing kernel particle method (RKPM) and methods based on radial basis functions (RBF). The reader may consult Refs. [5–9] for their definitions (in this paper we prefer to use the term “interpolation” for all the schemes although some of them do not have the Kronecker delta property). Several techniques such as collocation [10], Galerkin [7] or Petrov–Galerkin methods [11], etc., are also available for formulation of mesh-less methods. Variety of mesh-less methods are constructible by combining the available choices of the two parts (considering the limitations). For instance in the generalized finite difference (GFD) an LS together with a collocation method are used [5], in the finite point method (FPM) a WLS (or diffused MLS) is combined with a collocation method [10], in the element free Galerkin method (EFG) an MLS is used with a Galerkin type of formulation [7], and in the mesh-less local Petrov–Galerkin method (MLPG) again an MLS is combined with a Petrov–Galerkin formulation [11]. Other combinations are

---

B. Boroomand (✉) · M. Najjar  
Department of Civil Engineering, Isfahan University  
of Technology, Isfahan, Iran  
e-mail: boroomand@cc.iut.ac.ir

E. Oñate  
International Center for Numerical Methods in Engineering  
(CIMNE), UPC, Barcelona, Spain  
e-mail: onate@cimne.upc.edu

still possible (see the use of RKPM in a collocation approach in [12] for instance).

Regardless of the interpolation used, the mesh-less methods may be categorized in two main types, i.e. those which use integration on sub-domains [3, 7, 8, 11, 13–17] and those which do not [4, 9, 10, 12, 18–24]. The methods in the latter category are cheaper and faster for implementation when compared with those in the former category. However, there are some drawbacks such as loss of accuracy when using the second type. Nevertheless, due to the growing need for cheap and fast numerical methods, the second type is of interest of many scientists.

In this paper we present a method which on the one hand is based on using integration on sub-domains, and on the other hand the integrals are so simple and cheap that one can classify the method in the second category. Here we employ a weighted residual approach in a similar fashion as the sub-domain collocation method; however, the sub-domains we use here do not necessarily cover the whole domain. The sizes of the so defined sub-domains may be reduced, while the weight values are increased, so that the method becomes similar to a collocation method (similar to GFD and FPM). We shall use WLS (or diffused MLS) scheme for the approximation and thus the method we propose here can be considered as “generalized finite point method (GFPM)”.

By simplifying the integrals on the sub-domains, we shall present a library for the final results for each monomial term used in WLS. The final procedure thus becomes very simple and fast without the need for integrating through defining quadrature points. For sub-domains intersecting the boundaries the procedure is further modified to account for the effect of the boundary tractions.

As we shall show later in this paper, the decision on the sizes of the sub-domains plays an important role in the approximation. Therefore, along with the introduction of the method we shall propose a consistent procedure for finding a suitable size for each sub-domain. The procedure we propose here is in fact a generalized form of the patch test which is conventionally used for studies on the mesh-less methods. But unlike the conventional patch test in which a special combination of polynomial terms is used for the exact solution, here we consider all possible combinations of the polynomials which are of the same order with those used for the interpolation. Having ensured that GFPM generally passes the patch test, we shall proceed to find suitable sizes for the sub-domains by re-employing the patch test with polynomials of one order higher than those of the interpolation. This is performed by finding a minimum value for a suitable error norm through altering the sizes of the sub-domains in a fictitious domain defined as the patch test problem.

The content of the paper is arranged as follows. In Sect. 2 the procedure of WLS (or MLS) is revisited. In Sect. 3 we shall present the GFPM details. The library functions

obtained from the final results of the integrals used in GFPM will be presented in Sect. 3. Section 4 is devoted to explaining the procedure of the patch test and its generalized form. The results of the parameter tuning will be given in the same section. We shall present some numerical results in Sect. 5. Finally we summarize the conclusions in Sect. 6.

## 2 The WLS/MLS approximation

In this section we overview the procedure of the weighted least square method (WLS). The procedure of MLS has much in common with WLS, however, there are some differences between them which will be briefly explained. A generic function  $\hat{u}^c(\mathbf{x}^c)$ , with  $\mathbf{x}^c$  being a local coordinate system, is to be written locally in terms of a set of polynomials as

$$\hat{u}^c(\mathbf{x}^c) = \sum_{l=1}^m p_l(\mathbf{x}^c) a_l^c = \mathbf{p}^T(\mathbf{x}^c) \mathbf{a}^c \quad (1)$$

where  $\mathbf{p}(\mathbf{x}^c)$  is a vector of basis monomials and  $\mathbf{a}^c$  is a vector of coefficients. Superscript “c” is used to denote that the approximation is performed locally on a “cloud” area (see the cloud definition in Sect. 3.3). The coefficients  $\mathbf{a}^c$  are to be evaluated by a least square fit of  $\hat{u}^c(\mathbf{x}^c)$  over a set of data as  $\bar{u}_{0k}^c$  defined at  $N_p^c$  nodes ( $k = 1, \dots, N_p^c$ ) through using a suitable local weight function as  $w^c(\mathbf{x}^c)$  while  $\mathbf{x}^c$  is measured from a master node. Here “master node” refers to the node where the cloud is constructed and the local coordinate system is set up to define the approximate displacement field (see Sect. 3.3 for the cloud and the master node definitions).

The least square fit is performed by minimizing the error norm as

$$J^c = \sum_{k=1}^{N_p^c} w^c(\mathbf{x}_k^c) [\bar{u}_{0k}^c - \mathbf{p}^T(\mathbf{x}_k^c) \mathbf{a}^c]^2 \quad (2)$$

which leads to

$$\mathbf{a}^c = (\mathbf{A}^c)^{-1} \mathbf{B}^c \bar{\mathbf{u}}_0^c \quad (3)$$

with  $\bar{\mathbf{u}}_0^c$  being the vector of data and

$$\begin{aligned} \mathbf{A}^c &= \sum_{k=1}^{N_p^c} w^c(\mathbf{x}_k^c) \mathbf{p}(\mathbf{x}_k^c) \mathbf{p}^T(\mathbf{x}_k^c), \\ \mathbf{B}^c &= [w^c(\mathbf{x}_1^c) \mathbf{p}^c(\mathbf{x}_1^c), \dots, w^c(\mathbf{x}_{N_p^c}^c) \mathbf{p}^c(\mathbf{x}_{N_p^c}^c)] \end{aligned} \quad (4)$$

Then  $\hat{u}^c$  is evaluated as

$$\hat{u}^c(\mathbf{x}^c) = \mathbf{p}^T(\mathbf{x}^c) (\mathbf{A}^c)^{-1} \mathbf{B}^c \bar{\mathbf{u}}_0^c = \mathbf{N}_0^c(\mathbf{x}^c) \bar{\mathbf{u}}_0^c \quad (5)$$

In this paper we assume that  $\mathbf{A}^c$  in (3) is an invertible matrix and thus  $(\mathbf{A}^c)^{-1}$  is available. Obviously, reproduction of the data  $\bar{\mathbf{u}}_0^c$  by  $\hat{u}^c(\mathbf{x}^c)$  is not guaranteed since the shape functions do not have the Kronecker delta property and thus expression

(5) does not act as an interpolation. However, in this paper we shall use the term “interpolation” for convenience.

## 2.1 The difference between MLS and WLS

It is clear that the coefficients found in (3) are dependent on the location of the cloud. More specifically, if a pair of coordinates is assigned for the global location of the cloud, e.g.  $\mathbf{x}_0$  for the master node, the approximation used in (1) will then be dependent on  $\mathbf{x}_0$ , i.e.  $\hat{u}^c(\mathbf{x}^c, \mathbf{x}_0)$  and  $\mathbf{a}^c(\mathbf{x}_0)$ . This is essentially due to the fact that the weight function  $w^c(\mathbf{x}^c)$  used in (2) is defined locally and is a function of  $\mathbf{x}_0$  (and thus in a more precise form it should be written as  $w^c(\mathbf{x}^c, \mathbf{x}_0)$ ). Note also that in this sense the master node will not necessarily be a grid point (it can represent any chosen point in the domain). Considering such an effect, the approximation given in (5) will be a function of  $\mathbf{x}_0$  and may be written as

$$\hat{u}^c(\mathbf{x}^c, \mathbf{x}_0) = \mathbf{p}^T(\mathbf{x}^c) [\mathbf{A}^c(\mathbf{x}_0)]^{-1} \mathbf{B}^c(\mathbf{x}_0) \bar{\mathbf{u}}_0^c = \mathbf{N}_0^c(\mathbf{x}^c, \mathbf{x}_0) \bar{\mathbf{u}}_0^c \quad (6)$$

Expression (6) is sensitive to any movement of the cloud, i.e.  $d\mathbf{x}_0$ , and thus differentiation of  $\hat{u}^c(\mathbf{x}^c, \mathbf{x}_0)$  needs a sensitivity analysis for  $\mathbf{a}^c(\mathbf{x}_0)$  which should be performed

$$\bar{\mathbf{B}}^c = \begin{bmatrix} w(\mathbf{x}_1^c) \mathbf{p}(\mathbf{x}_1^c) & 0 & \dots & w(\mathbf{x}_2^c) \mathbf{p}(\mathbf{x}_2^c) & 0 & \dots \\ 0 & w(\mathbf{x}_1^c) \mathbf{p}(\mathbf{x}_1^c) & \dots & 0 & w(\mathbf{x}_2^c) \mathbf{p}(\mathbf{x}_2^c) & \dots \end{bmatrix}$$

by considering Eq. 5. Such a sensitivity analysis needs sensitivity of the matrix  $[\mathbf{A}^c(\mathbf{x}_0)]^{-1}$  which is a time consuming procedure. In this sense expression (6) is regarded as an MLS approximation. When MLS is to be used in a collocation method for the analysis of an elasticity problem, second derivatives of shape functions are needed and thus the procedure becomes even more complicated and time consuming. On the other hand when MLS is to be used in an integral-based mesh-less method, such as those introduced in [7, 8, 11] which just need first derivatives of shape functions, the procedure of cloud construction should be repeated for each integration point (again making the method expensive). In this paper we shall use “diffused derivatives” of (5) which means that we disregard the sensitivity of  $\mathbf{a}^c$  with respect to the location of the cloud. In this respect, (5) may be viewed as a simple WLS approximation/interpolation. Although we aim at using WLS in an integral-based scheme, as will be explained in the next sections, we shall just employ the cloud constructed at the master node (the cloud construction will not be repeated during the integration procedure). This makes the procedure very fast and cheap especially when compared with other integral-based mesh-less methods.

## 2.2 WLS in elasticity problems

In a two-dimensional elasticity problem the unknown field is of a vector form, so we shall use the result of (5) for each displacement component

$$\begin{aligned} \hat{\mathbf{u}}^c &= \mathbf{N}^c \bar{\mathbf{u}}^c \\ &= \sum_{k=1}^{N_p^c} \begin{bmatrix} \mathbf{p}^T(\mathbf{x}^c) (\mathbf{A}^c)^{-1} (\mathbf{B}^c)_k & 0 \\ 0 & \mathbf{p}^T(\mathbf{x}^c) (\mathbf{A}^c)^{-1} (\mathbf{B}^c)_k \end{bmatrix} \\ &\quad \times \begin{Bmatrix} \bar{u}_k^c \\ \bar{v}_k^c \end{Bmatrix} \end{aligned} \quad (7)$$

in which  $(\mathbf{B}^c)_k$  is the  $k$ th column of  $\mathbf{B}^c$  in (4). The above equation can be written as

$$\hat{\mathbf{u}}^c = \bar{\mathbf{P}}^c \bar{\mathbf{A}}^c \bar{\mathbf{B}}^c \bar{\mathbf{u}}^c, \quad \mathbf{N}^c = \bar{\mathbf{P}}^c \bar{\mathbf{A}}^c \bar{\mathbf{B}}^c \quad (8)$$

where

$$\bar{\mathbf{P}}^c = \begin{bmatrix} p_1(\mathbf{x}^c) & p_2(\mathbf{x}^c) & \dots & 0 & 0 & \dots \\ 0 & 0 & \dots & p_1(\mathbf{x}^c) & p_2(\mathbf{x}^c) & \dots \end{bmatrix},$$

$$\bar{\mathbf{A}}^c = \begin{bmatrix} (\mathbf{A}^c)^{-1} & 0 \\ 0 & (\mathbf{A}^c)^{-1} \end{bmatrix} \quad (9)$$

and

$$\bar{\mathbf{B}}^c = \begin{bmatrix} w(\mathbf{x}_{N_p^c}^c) \mathbf{p}(\mathbf{x}_{N_p^c}^c) & 0 & \dots & w(\mathbf{x}_{N_p^c}^c) \mathbf{p}(\mathbf{x}_{N_p^c}^c) & 0 & \dots \\ \dots & 0 & \dots & 0 & w(\mathbf{x}_{N_p^c}^c) \mathbf{p}(\mathbf{x}_{N_p^c}^c) & \dots \end{bmatrix} \quad (10)$$

Register for free at <https://www.scipedia.com> to download the version without the watermark

We shall use (7) when a set of library functions can be constructed in the new method.

## 3 The original and the generalized finite point method (FPM and GFPM)

We consider two-dimensional elasticity problems whose generic governing equation, in a domain of  $\Omega$ , is written as

$$\mathbf{S}^T \mathbf{D} \mathbf{S} \mathbf{u} + \mathbf{b} = \mathbf{0} \quad \text{in } \Omega \quad (11)$$

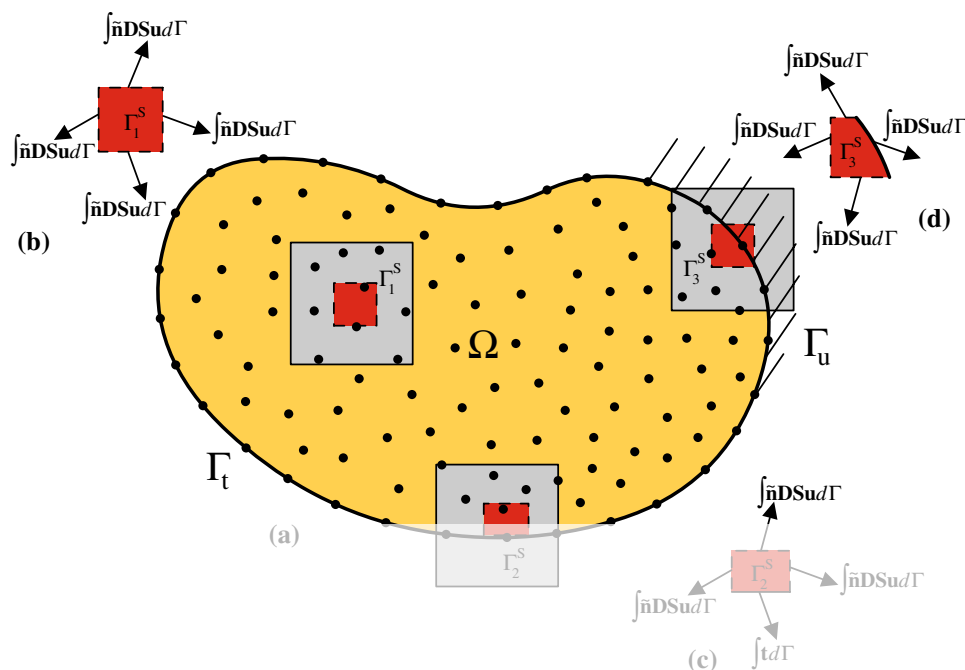
in which  $\mathbf{u}$  is a vector containing the displacement components,  $\mathbf{b}$  is the body force vector,  $\mathbf{D}$  is the matrix of material constants and  $\mathbf{S}$  is the well known matrix operator for defining the strains, i.e.  $\boldsymbol{\varepsilon} = \mathbf{S} \mathbf{u}$ . Equation 11 is to be solved with the following boundary conditions

$$\mathbf{u} = \mathbf{u}_B \quad \text{on } \Gamma_u \quad (12)$$

$$\hat{\mathbf{n}} \mathbf{D} \mathbf{S} \mathbf{u} = \mathbf{t} \quad \text{on } \Gamma_t \quad (13)$$

The whole boundary of the domain is partitioned as  $\partial\Omega = \Gamma_u \cup \Gamma_t$ . In (12) and (13), elements of  $\mathbf{u}_B$  are the prescribed

**Fig. 1** Schematic presentation of the discretization used for GFPM; **a** the main domain, **b** the interior sub-domains, **c** sub-domains intersecting Neumann boundaries, **d** sub-domains intersecting Dirichlet boundaries



displacements on boundary  $\Gamma_u$ , and  $\mathbf{t}$  is the vector of tractions defined on boundary  $\Gamma_t$ . Also it is an appropriate matrix containing components of the outward unit vector normal to the boundary.

In a weighted residual approach, Eq. 11 and the boundary conditions are written in a set of integral equations as

$$\int_{\Omega} \mathbf{v}_i^T (\mathbf{S}^T \mathbf{D} \mathbf{S} \mathbf{u} + \mathbf{b}) d\Omega + \int_{\Gamma_t} \alpha_i^T (\bar{\mathbf{n}} \mathbf{D} \mathbf{S} \mathbf{u} - \mathbf{t}) d\Gamma + \int_{\Gamma_u} \beta_i^T (\mathbf{u} - \mathbf{u}_B) d\Gamma = 0 \quad i = 1, 2, \dots \quad (14)$$

where for 2D problems

$$\mathbf{v}_i = v_i \mathbf{I}, \quad \alpha_i = \alpha_i \mathbf{I}, \quad \beta_i = \beta_i \mathbf{I} \quad (15)$$

in which  $\mathbf{I}$  is a  $2 \times 2$  identity matrix and,  $v_i, \alpha_i$  and  $\beta_i$  are weight/test functions and two sets of penalty functions, respectively. Next we discuss on the choices of the weight functions.

### 3.1 Discretization and formulation in FPM

In collocation methods, the domain  $\Omega$  and its boundary  $\partial\Omega$  are discretized into a set of points/nodes with coordinates as  $\mathbf{x}_i, i = 1, \dots, N_p$ . Then the weight and penalty parameters are chosen as Dirac delta functions, i.e.

$$v_i = A \delta(x - x_i) \delta(y - y_i), \quad \alpha_i = B \delta(x - x_i) \delta(y - y_i), \quad \beta_i = C \delta(x - x_i) \delta(y - y_i) \quad (16)$$

with  $A, B$  and  $C$  being some appropriate parameters. Substitution of (16) in (14) while using the approximated field of displacement in place of  $\mathbf{u}$ , leads to

$$A [\mathbf{S}^T \mathbf{D} \mathbf{S} \hat{\mathbf{u}}^c + \mathbf{b}]_{\mathbf{x}=\mathbf{x}_i} + B [\bar{\mathbf{n}} \mathbf{D} \mathbf{S} \hat{\mathbf{u}}^c - \mathbf{t}]_{\mathbf{x}=\mathbf{x}_i} + C [\hat{\mathbf{u}}^c - \mathbf{u}_B]_{\mathbf{x}=\mathbf{x}_i} = 0 \quad (17)$$

In FPM formulation we take  $B = C = 0$  for interior nodes,  $A = B = 0$  for nodes on  $\Gamma_u$ , and  $A = C = 0$  for nodes on  $\Gamma_t$ . In a modified form, for each node on  $\Gamma_t$  one may take just  $C = 0$  and  $A, B < 0$  where  $A$  and  $B$  are determined through the influence area of the node (see [4]).

### 3.2 Discretization and formulation in GFPM

As mentioned before, in GFPM we use a set of sub-domains, schematically shown in Fig. 1a, to discretize the  $\Omega$ . The sub-domains,  $\Omega_i^s$ , are constructed around nodes  $\mathbf{x}_i, i = 1, \dots, N_p$ . The union of such sub-domains does not necessarily cover the whole domain  $\cup_i \Omega_i^s \neq \Omega$ . The reader may note that the idea resembles that of FPM where the equilibrium equations are satisfied at a finite number of points rather than all points of the system (see also [11]). For each sub-domain we define a function as

$$W_i(x, y) = H\left(\frac{a}{2} - (x - x_i)\right) H\left(\frac{a}{2} + (x - x_i)\right) \times H\left(\frac{b}{2} - (y - y_i)\right) H\left(\frac{b}{2} + (y - y_i)\right) \quad (18)$$

where  $H(\cdot)$  is a Heaviside function and,  $a$  and  $b$  denote the sizes of the sub-domain (rectangular) along  $x$  and  $y$  axes.

*Case (a)* Sub-domains with no intersection with boundaries. In (14) and (15) we take  $v_i = AW_i$  and  $\alpha_i = 0, \beta_i = 0$ ,

$$A \int_{\Omega_i^s} (\mathbf{S}^T \mathbf{D} \mathbf{S} \hat{\mathbf{u}}^c + \mathbf{b}) d\Omega = \mathbf{0} \quad (19)$$

$$\mathbf{x}_i \in \Omega \wedge \Omega_i^s \subset \Omega \wedge \Gamma_i^s \cap \partial\Omega = \emptyset$$

with  $A$  being an arbitrary coefficient, we have

$$\int_{\Omega_i^s} (\mathbf{S}^T \mathbf{D} \mathbf{S} \hat{\mathbf{u}}^c + \mathbf{b}) d\Omega = \mathbf{0} \quad (20)$$

which can be simplified with the use of integration by parts as

$$\int_{\Gamma_i^s} \tilde{\mathbf{n}} \mathbf{D} \mathbf{S} \hat{\mathbf{u}}^c d\Gamma + \int_{\Omega_i^s} \mathbf{b} d\Omega = \mathbf{0} \quad \text{or} \quad (21)$$

$$\int_{\Gamma_i^s} \tilde{\mathbf{n}} \mathbf{D} \mathbf{S} \hat{\mathbf{u}}^c d\Gamma = - \int_{\Omega_i^s} \mathbf{b} d\Omega$$

where  $\Gamma_i^s$  is the boundary of the  $i$ th sub-domain. Equation 21 represents the equilibrium state of the sub-domain when the approximate tractions together with body forces are present (see Fig. 1b). Now substitution of (8) in (21) leads to

$$\int_{\Gamma_i^s} \tilde{\mathbf{n}} \mathbf{D} \mathbf{S} \bar{\mathbf{A}}^c \bar{\mathbf{B}}^c \bar{\mathbf{u}}^c d\Gamma = - \int_{\Omega_i^s} \mathbf{b} d\Omega \quad (22)$$

$$\text{or} \quad \mathbf{H}_i \bar{\mathbf{u}}^c = \mathbf{f}_i, \quad \mathbf{H}_i = \int_{\Gamma_i^s} \tilde{\mathbf{n}} \mathbf{D} \mathbf{S} \bar{\mathbf{A}}^c \bar{\mathbf{B}}^c d\Gamma = \bar{\mathbf{G}}^c \bar{\mathbf{A}}^c \bar{\mathbf{B}}^c,$$

$$\bar{\mathbf{G}}^c = \int_{\Gamma_i^s} \tilde{\mathbf{n}} \mathbf{D} \mathbf{S} \bar{\mathbf{A}}^c d\Gamma, \quad \mathbf{f}_i = - \int_{\Omega_i^s} \mathbf{b} d\Omega$$

The so defined matrix  $\bar{\mathbf{G}}^c$  can be explicitly evaluated when the sub-domain is of a regular shape, like a rectangle, etc. The dimension of the matrix is dependent on the number of polynomial terms. The explicit form of  $\bar{\mathbf{G}}^c$  is given in the “Appendix”. Another point to note is that the order of differentiation of shape functions needed in GFPM is less than that of FPM [compare (17) with (22)]. Moreover, in the case of presence of point loads Eq. 22 takes the load into account while it is not the case for Eq. 17.

*Remark* The reader may note that the weighted form written as (19) has much in common with the well known Petrov–Galerkin method using weight functions as defined in (18). It can be shown that the collocation approach used in FPM is in fact a special case of (19). To this end one may rewrite (19) as

$$\int_{\Omega} (\mathbf{S}^T \mathbf{D} \mathbf{S} \hat{\mathbf{u}}^c + \mathbf{b})(AW_i) d\Omega = \mathbf{0} \quad (24)$$

while requiring that

$$\int_{\Omega} (AW_i) d\Omega = 1 \quad \text{or} \quad A \int_{\Omega_i^s} d\Omega = 1 \quad (25)$$

which leads to

$$A = \frac{1}{\Omega_i^s}, \quad \Omega_i^s = ab \quad (26)$$

Now by inserting (26) in (24) and reducing the sub-domain size so that  $\Omega_i^s \rightarrow 0$ , the following equation is resulted

$$\int_{\Omega_i^s} (\mathbf{S}^T \mathbf{D} \mathbf{S} \hat{\mathbf{u}}^c + \mathbf{b}) \delta(\mathbf{x} - \mathbf{x}_i) d\Omega = \mathbf{0}, \quad (27)$$

$$\delta(\mathbf{x} - \mathbf{x}_i) = \delta(x - x_i) \delta(y - y_i)$$

Which is equivalent to

$$[\mathbf{S}^T \mathbf{D} \mathbf{S} \hat{\mathbf{u}}^c + \mathbf{b}]_{\mathbf{x}=\mathbf{x}_i} = \mathbf{0} \quad (28)$$

The above equation is exactly the same as the one used in FPM formulation for interior points. Therefore, the method we present here in this paper may be regarded as the generalized form of FPM. Similar discussion may be made for other cases of point locations, Cases (b) and (c) coming next. The final expressions for such cases will be similar to those given in Ref. [4] for FPM.

*Case (b)* Sub-domains intersecting boundaries with their master node not falling on  $\Gamma_u$ . In this case we take  $v_i = AW_i$  and  $\alpha_i = BW_i, \beta_i = 0$

$$\int_{\Omega_i^s} (\mathbf{S}^T \mathbf{D} \mathbf{S} \hat{\mathbf{u}}^c + \mathbf{b}) d\Omega + \int_{\Gamma_i^s \cap \Gamma_t} (\tilde{\mathbf{n}} \mathbf{D} \mathbf{S} \hat{\mathbf{u}}^c - \mathbf{t}) d\Gamma = \mathbf{0} \quad (29)$$

$$\mathbf{x}_i \notin \Gamma_u \wedge \Gamma_i^s \cap \partial\Omega \neq \emptyset \quad (29)$$

Considering  $B = -A$  [4] with  $A$  being an arbitrary coefficient we have

$$\int_{\Omega_i^s} (\mathbf{S}^T \mathbf{D} \mathbf{S} \hat{\mathbf{u}}^c + \mathbf{b}) d\Omega - \int_{\Gamma_i^s \cap \Gamma_t} (\tilde{\mathbf{n}} \mathbf{D} \mathbf{S} \hat{\mathbf{u}}^c - \mathbf{t}) d\Gamma = \mathbf{0} \quad (30)$$

or

$$\int_{\Gamma_i^s} \tilde{\mathbf{n}} \mathbf{D} \mathbf{S} \hat{\mathbf{u}}^c d\Gamma + \int_{\Omega_i^s} \mathbf{b} d\Omega - \int_{\Gamma_i^s \cap \Gamma_t} (\tilde{\mathbf{n}} \mathbf{D} \mathbf{S} \hat{\mathbf{u}}^c) d\Gamma + \int_{\Gamma_i^s \cap \Gamma_t} \mathbf{t} d\Gamma = \mathbf{0} \quad (31)$$

which can be rewritten as

$$(\mathbf{H}_i - \mathbf{H}_i') \bar{\mathbf{u}}^c = \mathbf{f}_i, \quad \mathbf{H}_i' = \int_{\Gamma_i^s \cap \Gamma_t} \tilde{\mathbf{n}} \mathbf{D} \mathbf{S} \bar{\mathbf{N}}^c d\Gamma, \quad (32)$$

$$\mathbf{f}_i = - \int_{\Gamma_i^s \cap \Gamma_t} \mathbf{t} d\Gamma - \int_{\Omega_i^s} \mathbf{b} d\Omega$$

Register for free at <https://www.scipedia.com> to download the version without the watermark



Equation 31 represents the equilibrium state of the sub-domain with approximate tractions in one part and true tractions at another part (see Fig. 1c). Here the reader may note that near corners with tractions prescribed at two faces with different unit normal vectors, no ambiguity is made for the user while working with (32), but it is not the case for (17).

In evaluation of  $\mathbf{H}'_i$  in (32) the integration is performed on the segments of the boundary. When the sub-domain is of regular shape, the segments are of the form of small lines. The integrals are then taken along the lines defined by their two end points (a numerical integration may be needed if the boundary of the sub-domain is a curve but when the sub-domain is very small the boundaries can be approximated by lines). By the two end points one can define the unit normal to the boundary needed in the matrix  $\tilde{\mathbf{n}}$ . The final results of such integrations, in terms of the coordinates of the two end points, are given in the “Appendix”.

*Case (c)* Sub-domains intersecting boundaries with their master node falling on  $\Gamma_u$ . In this case we take  $v_i = AW_i$ ,  $\alpha_i = BW_i$ ,  $\beta_i = C\delta(x - x_i)\delta(y - y_i)$

$$A \int_{\Omega^s_i} (\mathbf{S}^T \mathbf{D} \mathbf{S} \hat{\mathbf{u}}^c + \mathbf{b}) d\Omega + B \int_{\Gamma^s_i \cap \Gamma_u} (\tilde{\mathbf{n}} \mathbf{D} \mathbf{S} \hat{\mathbf{u}}^c - \tilde{\mathbf{t}}) d\Gamma + C [\hat{\mathbf{u}}^c - \mathbf{u}_B]_{\mathbf{x}=\mathbf{x}_i} = \mathbf{0} \quad (33)$$

$\mathbf{x}_i \in \Gamma_u \wedge \Gamma^s_i \cap \partial\Omega \neq \emptyset$

Considering  $B = -A$  and  $(A, C)$  as a pair of arbitrary coefficients, we have

$$\int_{\Omega^s_i} (\mathbf{S}^T \mathbf{D} \mathbf{S} \hat{\mathbf{u}}^c + \mathbf{b}) d\Omega - \int_{\Gamma^s_i \cap \Gamma_u} (\tilde{\mathbf{n}} \mathbf{D} \mathbf{S} \hat{\mathbf{u}}^c - \tilde{\mathbf{t}}) d\Gamma = \mathbf{0} \quad (34)$$

$$[\hat{\mathbf{u}}^c - \mathbf{u}_B]_{\mathbf{x}=\mathbf{x}_i} = \mathbf{0} \quad (35)$$

which leads to

$$(\mathbf{H}_i - \mathbf{H}'_i) \hat{\mathbf{u}}^c = \mathbf{f}_i, \quad \mathbf{f}_i = -\mathbf{f}^{\tilde{t}}_i - \int_{\Omega^s_i} \mathbf{b} d\Omega, \quad \mathbf{f}^{\tilde{t}}_i = \int_{\Gamma^s_i \cap \Gamma_u} \tilde{\mathbf{t}} d\Gamma \quad (36)$$

$$[\hat{\mathbf{u}}^c]_{\mathbf{x}=\mathbf{x}_i} = [\mathbf{u}_B]_{\mathbf{x}=\mathbf{x}_i} \quad (37)$$

with  $\mathbf{H}'_i$  defined in (32) and  $\tilde{\mathbf{t}}$  being the unknown tractions at the boundary  $\Gamma_u$ . The coefficient matrix for the whole domain can now be arranged by considering (23), (32) and (36) for  $i = 1, \dots, N_p$  as

$$\mathbf{K} \hat{\mathbf{u}} = \mathbf{F} \quad (38)$$

where  $\mathbf{K}$  is  $2N_p \times 2N_p$  matrix. Note that  $\mathbf{f}^{\tilde{t}}_i$  is a set of unknowns while  $[\mathbf{u}_B]_{\mathbf{x}=\mathbf{x}_i}$  is a set of known values. Therefore, the solution procedure of (38) becomes similar to its counterpart in FEM solution for instance.

### 3.3 The cloud and sub-domain definitions

To illustrate the essential features of the main formulation, in this paper we shall follow a simple cloud definition for both FPM and GFPM. Having selected a grid node, here known as the master node, the cloud is constructed by sequential enlargement of a square, with edges parallel to the global axes and centered at the master node, so that a minimum number nodes fall in the square. The distances between the master node and the most remote nodes along the global axes are determined by calculating  $|\Delta x|_{\max}$  and  $|\Delta y|_{\max}$ .

In order to perform the WLS procedure, one may use a local normalized coordinates as  $\xi = \frac{(x-x_i)}{|\Delta x|_{\max}}$ ,  $\eta = \frac{(y-y_i)}{|\Delta y|_{\max}}$ , with  $(x_i, y_i)$  being the coordinates of the master node. Such a normalization affects the regularity of the matrix  $\mathbf{A}$  in (3) and (4), see [4] for instance. Having evaluated the shape functions in terms of the normalized coordinates, an inverse mapping can be used to express the shape functions in term of the global coordinates (also for derivatives of them).

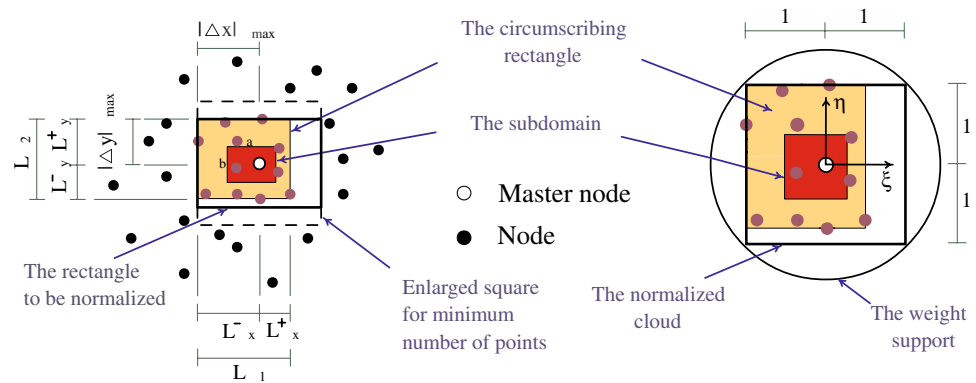
For sub-domain definition, we first define the circumscribing rectangle of the nodes falling in the cloud. The edge sizes of such circumscribing rectangle are found from the distances of the most remote nodes at either sides of the master node along each axes, say  $L_1 = |L^+_x| + |L^-_x|$  and  $L_2 = |L^+_y| + |L^-_y|$  (see Fig. 2). The sub-domain is defined by considering a rectangle containing the master node with its edges being proportional to those of the circumscribing rectangle,  $a = \bar{a}L_1$ ,  $b = \bar{b}L_2$  with  $0 \leq \bar{a} \leq 1$  and  $0 \leq \bar{b} \leq 1$ .

Register for free at <https://www.scipedia.com> to download the version without the watermark

#### 4 The patch test and its generalized form

It is important to show that the GFPM can pass some standard patch tests. The patch test is usually designed by selecting a small domain and choosing an exact field expressed in terms of polynomials with the same order as (or smaller order than) those used in the interpolation (in this case those used in WLS). The boundary conditions are determined from the exact solution (either in the form of Dirichlet or Neumann form). It is expected that the numerical solution fully reproduce the exact field.

In this paper we generalize the patch test procedure to consider all possible polynomial sets, with arbitrary coefficients. This will be performed in two forms. The first one is a form in which the order of the monomials is chosen similar to that of the interpolation functions. We recognize this form as “the conventional patch test in its extended form”. The second one is designed so that the order of the monomials is chosen to be of one order higher than that of the interpolation. This latter form is recognized as “the generalized patch test” and is to be used for optimizing the size of the sub-domain.

**Fig. 2** The definition of the cloud and the sub-domain used

#### 4.1 Conventional patch test in its extended form

Suppose that the polynomial used in the WLS are of  $m$  order, a displacement field is assumed as a summation of monomials with highest order equal to  $m$  as

$$\mathbf{u}^{ex} = \tilde{\mathbf{P}}\tilde{\mathbf{a}} = \sum_{i,j} \tilde{\mathbf{P}}_{i,j}^u \tilde{a}_{i,j}^u + \sum_{k,l} \tilde{\mathbf{P}}_{k,l}^v \tilde{a}_{k,l}^v, \quad (39)$$

$$\tilde{\mathbf{P}}_{i,j}^u = \begin{Bmatrix} x^i y^j \\ 0 \end{Bmatrix}, \quad \tilde{\mathbf{P}}_{k,l}^v = \begin{Bmatrix} 0 \\ x^k y^l \end{Bmatrix}$$

In (39)  $\tilde{\mathbf{P}}_{i,j}^u$  and  $\tilde{\mathbf{P}}_{k,l}^v$  represent the columns of  $\tilde{\mathbf{P}}$ , containing monomials for  $u$  and  $v$  displacements, and  $\tilde{a}_{i,j}^u$  ( $\tilde{a}_{k,l}^v$ ) are the associated coefficients arranged in  $\tilde{\mathbf{a}}$ . Now a set of new exact fields are considered as

$$\mathbf{u}_{\tilde{n}}^{ex} = \tilde{\mathbf{P}}_{\tilde{n}} \tilde{\mathbf{a}}_{\tilde{n}}, \quad \tilde{n} = 1, \dots, 2m \quad (40)$$

and a set of problems with so defined exact solutions are solved on a patch domain as  $\Omega_p$  [in (40)  $\tilde{\mathbf{P}}_{\tilde{n}}$  is a generic column of  $\tilde{\mathbf{P}}$ ]. To this end, for each numerical solution the information from the associated exact solution is used as the tractions or the prescribed boundary conditions on  $\partial\Omega_p$ . Note that whenever is needed we evaluate the body forces due to the exact solution as  $\mathbf{b}_{\tilde{n}} = -\mathbf{S}^T \mathbf{D} \mathbf{S} \mathbf{u}_{\tilde{n}}^{ex}$ . Therefore, a set of solutions, according to Sect. 4, are performed on  $\Omega_p$  to find nodal values of the displacements as

$$\tilde{\mathbf{u}}_{\tilde{n}} = \mathbf{K}^{-1} \mathbf{F}_{\tilde{n}} \quad (41)$$

In above  $\tilde{\mathbf{u}}_{\tilde{n}}$  contains the nodal displacement values when  $\tilde{\mathbf{P}}_{\tilde{n}}$  (i.e.  $\tilde{\mathbf{P}}_{i,j}^u$  or  $\tilde{\mathbf{P}}_{k,l}^v$ ) plays the role of the exact solution. In that case  $\mathbf{F}_{\tilde{n}}$  is a vector similar to the right hand side of (38) and is associated with the tractions and the body forces induced by  $\tilde{\mathbf{P}}_{\tilde{n}}$ .

The point-wise errors can now be evaluated as

$$\mathbf{e}_{\tilde{n}} = \tilde{\mathbf{u}}_{\tilde{n}}^{ex} - \tilde{\mathbf{u}}_{\tilde{n}} = \tilde{\mathbf{u}}_{\tilde{n}}^{ex} - \mathbf{K}^{-1} \mathbf{F}_{\tilde{n}} \quad (42)$$

Here  $\tilde{\mathbf{u}}_{\tilde{n}}^{ex}$  is the exact nodal displacement values evaluated from  $\tilde{\mathbf{P}}_{\tilde{n}}$  ( $\tilde{\mathbf{P}}_{i,j}^u$  or  $\tilde{\mathbf{P}}_{k,l}^v$ ) and arranged with the same order as

the ones in  $\tilde{\mathbf{u}}$ . Now considering the unknown coefficients, one may write

$$\mathbf{e} = \sum_{\tilde{n}=1}^{2m} \mathbf{e}_{\tilde{n}} \tilde{a}_{\tilde{n}} = \mathbf{E} \tilde{\mathbf{a}}, \quad \mathbf{E} = [\mathbf{e}_1 \mathbf{e}_2 \dots \mathbf{e}_{2m}] \quad (43)$$

An Euclidean norm of displacement errors (conveniently in a squared form) can now be evaluated

$$(\mathbf{e})^2 = \frac{1}{N_p} (\mathbf{e})^T \mathbf{e} = \frac{1}{N_p} \tilde{\mathbf{a}}^T \mathbf{Q} \tilde{\mathbf{a}}, \quad \mathbf{Q} = (\mathbf{E})^T \mathbf{E} \quad (44)$$

The error analysis may be performed by finding the maximum eigenvalue of  $\mathbf{Q}$ , which represents the  $L_2$  norm of the matrix, so that

$$|\mathbf{Q} - \lambda_{\max} \mathbf{I}| = 0 \quad (45)$$

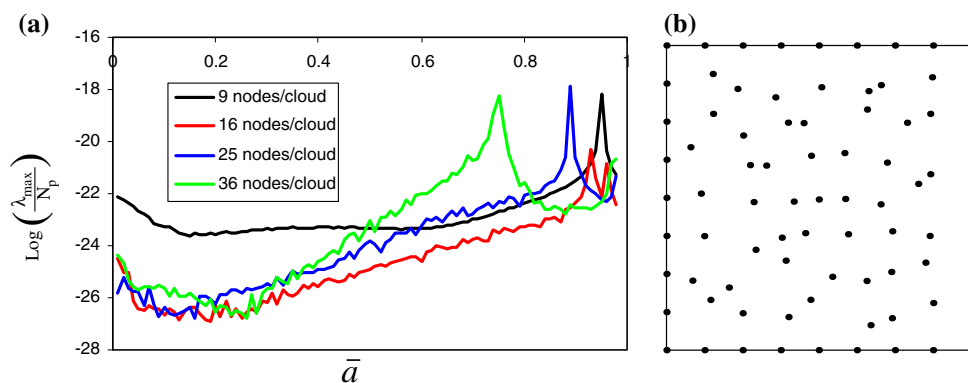
Register for free at <https://www.scipedia.com> to download the version without the watermark

that the maximum eigenvalue vanishes, the error norm will be zero for all possible coefficients  $\tilde{\mathbf{a}}$  and thus the patch test will be passed. One of the advantages of the so defined patch test is that when the patch test is not fully passed, to obtain an insight to the problem, the user can determine the combination of the polynomial terms which gives  $\lambda_{\max}$  by evaluating the associated eigenvector of  $\mathbf{Q}$ .

##### 4.1.1 The results of the conventional patch test in its extended form

In this section we present the results of the tests for a generic problem with specific properties. Consider a generic 2D plane stress elasticity problem; it can be observed that the patch test formulation is not sensitive to the Young's modulus of the material. However, this is not the case for Poisson's ratio. Therefore we consider a problem with Young's modulus  $E=1$  and a set of Poisson's ratios as  $\nu \in \{0.05 \ 0.15 \ 0.3 \ 0.45\}$  for instance. To begin the test, we choose a complete second order polynomial for interpolation and a Gaussian weight function as

**Fig. 3** Effect of round-off errors in patch tests ( $\nu = 0.3$ ,  $N_p = 81$ ) with irregular grids; **a** variation of  $\lambda_{\max}/N_p$  in logarithmic scale versus  $\bar{a}$ , **b** the grid used



$$w(r) = \begin{cases} \frac{\exp(-r^2/r_c^2) - \exp(-r_m^2/r_c^2)}{1 - \exp(-r_m^2/r_c^2)} & 0 \leq r \leq r_m \\ 0 & r > r_m \end{cases} \quad (46)$$

In (46),  $r = \sqrt{\xi^2 + \eta^2}$  with  $\xi$  and  $\eta$  being the normalized coordinates (see Sect. 3.3 for the cloud definition) also  $r_c$  and  $r_m$  denote two distances proportional to the sizes of the cloud. In this paper we have used  $r_c = 0.35$  and  $r_m = 1.4$  noting that in the normalized coordinates the cloud is a  $2 \times 2$  square. The size of the sub-domain may be expressed by a non-dimensional parameter as  $\bar{a} = \frac{a}{L_1}$  and  $\bar{b} = \frac{b}{L_2}$ , with  $L_1$  and  $L_2$  being the sizes of the circumscribing rectangle (see Fig. 2). In the tests we consider  $\bar{a} = \bar{b}$ .

We consider a set of  $2 \times 2$  domains with  $5 \times 5$ ,  $9 \times 9$ ,  $11 \times 11$  and  $21 \times 21$  grids of nodes with regular distribution. The result of the patch test, for all the grids, is as

$$\lambda_{\max} = 0 \quad \forall \nu \in \{0.05 \ 0.15 \ 0.3 \ 0.45\} \wedge \quad \forall \bar{a} \in \{0.1 \ 0.2 \ \dots \ 0.9 \ 1.0\} \quad (47)$$

for second and third order complete polynomials (and for different numbers of nodes in each cloud chosen as 9, 16, 25 and 36). The above result is valid for both types of boundary conditions, i.e. Dirichlet and Neumann types. For the latter type we have used information from the exact displacement field to provide minimum essential boundary conditions. The same result is obtained for irregular grids, constructed by perturbation of the grid points (see also Sect. 4.3.1 for irregularity index and the grid used). It may be expected that the use of irregular grids introduce some round-off errors in the computations. However, our experience for this case shows that such errors are at the level of machine precision. To give an insight to such an effect, here we report the values obtained for  $\text{Log}(\lambda_{\max}/N_p)$  in Fig. 3a for an irregular grid shown in Fig. 3b. In the tests a complete second order polynomial has been used for the interpolation. Similar results are obtained when a third order polynomial is employed.

## 4.2 Generalized patch test and parameter tuning

In this section we generalize the patch test presented in the preceding section. Here we use a polynomial displacement field with one order higher than the one employed in WLS, assuming that asymptotically, i.e. when the grid of nodes becomes very fine, the error of computation pertains to the corresponding terms in the Taylor's series of the exact solution (see [25,26] for patch tests in error analyses in finite element method).

Obviously it is not expected that the GFPM reproduce the exact field. The aim is to optimize the size of the sub-domains so that the errors become as small as possible. The procedure of error evaluation is similar to what explained in (39)–(44). The statement of the optimization procedure may be given as follows

$$\text{Find } (\bar{a}, \bar{b}) \quad \bar{a} = \frac{a}{L_1}, \bar{b} = \frac{b}{L_2}$$

To minimize  $\lambda_{\max}$

$$\text{so that } |\mathbf{Q} - \lambda_{\max} \mathbf{I}| = 0 \quad (48)$$

$$\text{with side constraints } \begin{cases} 0 < \bar{a} \leq 1 \\ 0 < \bar{b} \leq 1 \end{cases}$$

The minimization is simple to perform and there is no need for using sophisticated mathematical programming. Here we find the optimal values of  $(\bar{a}, \bar{b})$  by evaluation of  $\lambda_{\max}$  on a fine grid defined on the feasible domain  $0 < \bar{a} \leq 1, 0 < \bar{b} \leq 1$ .

### 4.2.1 The optimization procedure in its step by step form

The step by step procedure is as follows:

1. Choose a patch domain with a number of points/nodes (as the solution domain).
2. Choose a complete polynomial for construction of shape functions (with order of  $m$  for instance).
3. Consider another complete polynomial with one order (or more) higher than that of shape functions (with  $m + 1$  order for instance).



4. Choose a pair of  $(\bar{a}, \bar{b})$  from a series selected so that  $0 < \bar{a} \leq 1, 0 < \bar{b} \leq 1$ .
5. Construct  $\mathbf{K}$  matrix, as defined in (38), for the numerical solution.
6. Choose a vector of  $\tilde{\mathbf{P}}_{\bar{n}}(\tilde{\mathbf{P}}_{i,j}^u$  or  $\tilde{\mathbf{P}}_{k,l}^v)$  from the series as defined in (39) considering the polynomial chosen in step 3, i.e.  $i + j \leq m + 1$  or  $k + l \leq m + 1$ .
7. Define the right hand side vector  $\mathbf{F}_{\bar{n}}$  from the boundary tractions/displacements associated with  $\tilde{\mathbf{P}}_{\bar{n}}$  as the exact solution (find the associated  $\mathbf{b}_{\bar{n}}$  if needed).
8. Solve Eq. 41 for  $\bar{\mathbf{u}}_{\bar{n}}$ .
9. Find the nodal values of the exact solution.
10. Construct the error vector  $\mathbf{e}_{\bar{n}}$  as defined in (42).
11. Repeat from step 6 to construct  $\mathbf{E}$  as defined in (43).
12. Construct  $\mathbf{Q}$  as defined in (44).
13. Perform an eigenvalue solution  $[\mathbf{Q} - \lambda \mathbf{I}] = 0$  to find the maximum eigenvalue  $\lambda_{\max}$  ( $\lambda_{\max}$  may be stored in a vector as  $\lambda$  for further use).
14. Repeat from step 4.
15. Find the minimum element of  $\lambda$  associate with the optimal pair  $(\bar{a}_{\text{opt}}, \bar{b}_{\text{opt}})$ .

The reader may note that the above procedure is followed for a generic problem with specific polynomial order, cloud definition, etc. The optimization may be performed for a set of problems and the results can be tabulated for general use.

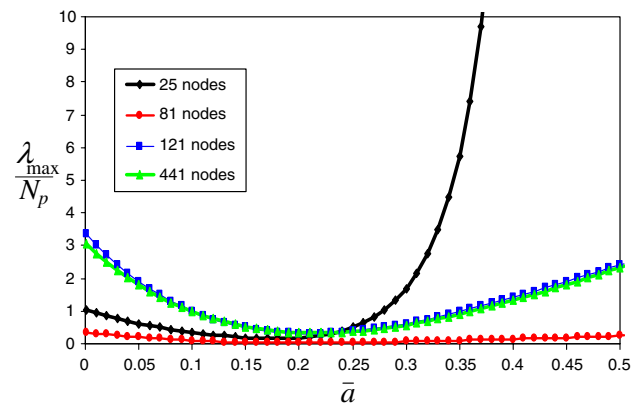
#### 4.3 The results of the optimization

In this section we present the results of the optimization procedure for some generic problems with specific properties. As before we consider a problem with Young's modulus  $E = 1$ , and to begin with, we choose Poisson's ratio  $\nu = 0.3$  (the effects of considering other values for Poisson's ratio will be given later). Also in the optimization procedure we consider  $\bar{a} = \bar{b}$ .

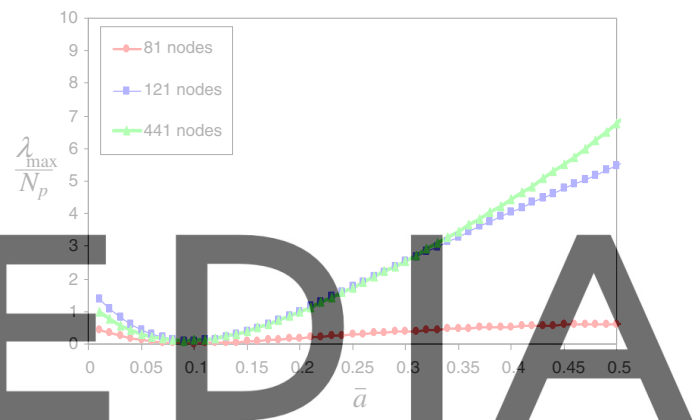
We shall give the results of the optimization for domains with regular node distribution and then take into account the irregularity effects by a correction factor obtained for a set of domains with irregular node distribution.

##### 4.3.1 Domains with regular node distribution

Here again we consider a set of  $2 \times 2$  domains with grid of  $5 \times 5, 9 \times 9, 11 \times 11$  and  $21 \times 21$  equally spaced nodes along the two global axes  $x$  and  $y$ . Figure 4 depicts the variation of the  $\lambda_{\max}$  with respect to the size of the sub-domain,  $\bar{a}$  (note that  $\bar{a} = \bar{b}$  in this paper). In this figure the minimum number of nodes in each cloud is 9. As is seen in the figure an optimal value can be found for each series of patch tests. It may be observed that an interval of  $0.1 \leq \bar{a}_{\text{opt}} \leq 0.3$  contains all optimal values. This clearly shows that it is not necessary to consider sub-domains covering the whole domain and in



**Fig. 4** Variation of  $\lambda_{\max}/N_p$  versus  $\bar{a}$ , for problems with  $\nu = 0.3$  and with minimum 9 nodes in each cloud. A complete second order polynomial has been used for the interpolation

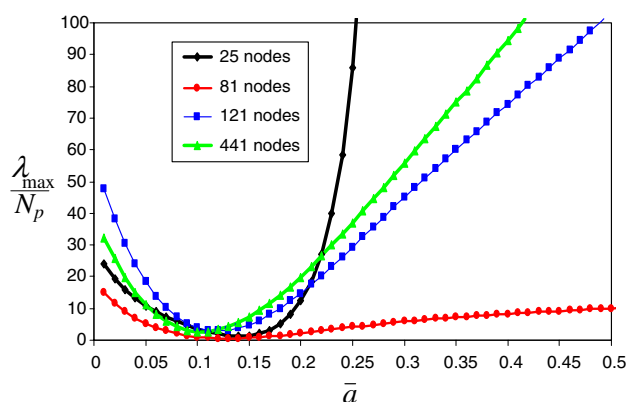


**Fig. 5** Variation of  $\lambda_{\max}/N_p$  versus  $\bar{a}$ , for problems with  $\nu = 0.3$  and with minimum 25 nodes in each cloud. A complete second order polynomial has been used for the interpolation

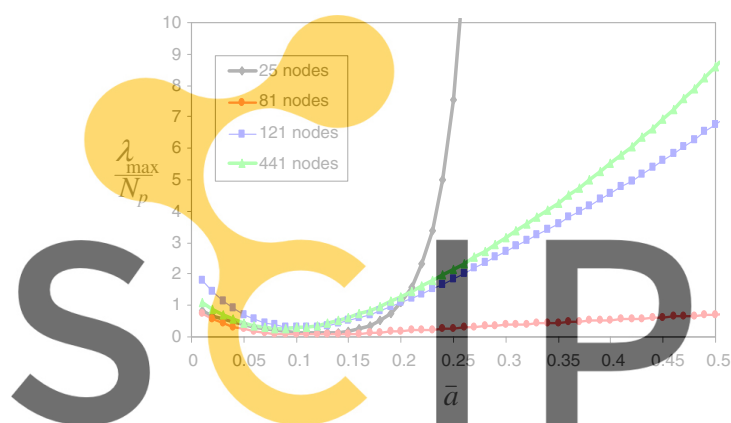
fact non-overlapping sub-domains are more appropriate. The effect of the minimum number of nodes in each cloud can be seen in Fig. 5. The figure shows that the optimal values of  $\bar{a}$  are slightly less than those in Fig. 4. The interval of  $0.05 \leq \bar{a}_{\text{opt}} \leq 0.15$  is recognized for optimal values in the figure.

It may be interesting to know what happens if one chooses a polynomial of two-order higher than that used in the interpolation. For instance in this case we examine the use of a fourth order polynomial in the optimization procedure. The results are shown in Fig. 6 when at least 16 nodes are selected in each cloud. It can be seen that the optimal intervals are nearly the same as those obtained for a third order polynomial. We do not consider it necessary to use such polynomial in this study.

As mentioned earlier, Poisson's ratio might still play a prominent role in the optimization. To study the effects, we have repeated the procedure for different Poisson's ratios. Figure 7 depicts a set of results obtained for  $\nu = 0.45$  and when 16 nodes are considered in each cloud. Based on the



**Fig. 6** The results of the optimization when a fourth order polynomial is used. Variation of  $\lambda_{\max}/N_p$  versus  $\bar{a}$ , for problems with  $\nu = 0.3$  and with minimum 16 nodes in each cloud. A complete second order polynomial has been used for the interpolation



**Fig. 7** Variation of  $\lambda_{\max}/N_p$  versus  $\bar{a}$ , for problems with  $\nu = 0.45$  and with minimum 16 nodes in each cloud. A complete second order polynomial has been used for the interpolation

observations on several optimization processes, we suggest the optimal values for sub-domain sizes in Table 1.

In order to use the table for other cases not shown, an interpolation/extrapolation may be used. The whole procedure may be repeated for when a complete third order polynomial is employed for interpolation. In that case a complete fourth order polynomial is used for parameter tuning. The results, identical to the first set, are reported in the same table.

The reader may note that the suggested optimal sub-domain sizes are essentially useful for problems with smooth solutions since in the optimization procedure just polynomial terms are used as the exact solution. However, as will be seen in the section of numerical examples, the method works reasonably well for the case in which a reentrant corner exists (fourth example). Nevertheless, a similar optimization procedure may be followed by the use of singular functions, as the exact solution, to find optimal sub-domain sizes around points where singularity is expected in the solution. Such a study is beyond the scopes of this paper and will be presented in forthcoming papers.

**Table 1** Suggested values for sub-domain size in clouds with different number of nodes and with a complete quadratic/cubic polynomial for interpolation

The number of nodes in the cloud $\nu \in \{0.05 \quad 0.15 \quad 0.3 \quad 0.45\}$			
9 nodes (Quad. only)	16 nodes	25 nodes	36 nodes
0.12	0.12	0.12	0.35

#### 4.3.2 The effect of irregularity of the node distribution

The results we have presented so far are mainly suitable for regular node distributions. In this section we study the effect of irregularity. However, we should make a tool for measuring the irregularity. To this end we first define an irregularity index as

$$S = \sqrt{\frac{I_{\xi\eta}^0}{N_p^c}} \quad I_{\xi\eta}^0 = \sum_{k=1}^{N_p^c} (\xi_k - \xi_0)(\eta_k - \eta_0) \quad (49)$$

$$\xi_0 = \frac{1}{N_p^c} \sum_{k=1}^{N_p^c} \xi_k, \quad \eta_0 = \frac{1}{N_p^c} \sum_{k=1}^{N_p^c} \eta_k$$

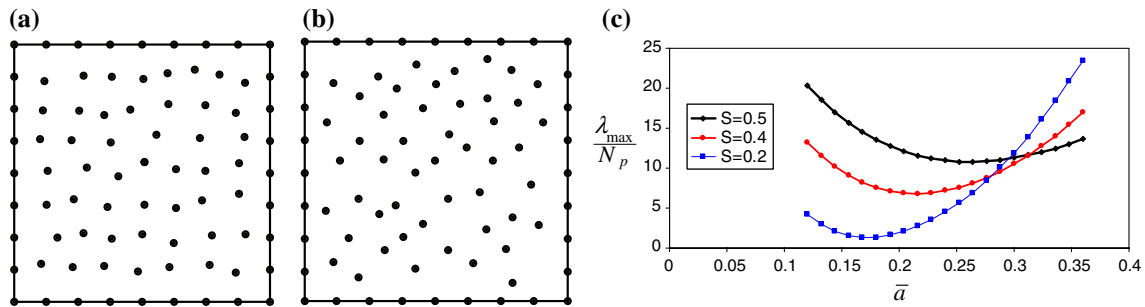
In above  $I_{\xi\eta}^0$  represents the product inertia of the nodes about the local normalized axes passing through the mass center of the nodes. In the second step we have constructed some patches with irregular node distribution. To construct each patch, the coordinates of the nodes in a  $9 \times 9$  regular patch have been randomly perturbed. The magnitude of the perturbation has been altered to obtain patches with different irregularity (see Fig. 8c, for two sample patches for instance).

It has been assumed that the behavior of the patch is dependent on the maximum irregularity index calculated in all clouds of the patch. Therefore, a unique irregularity index can be allocated to each patch of nodes. The optimization procedure is then repeated for different irregularity index. Figure 8c depicts the variation of  $\lambda_{\max}/N_p$  in terms of  $\bar{a}$  for some sampling index.

The figure shows that the location of optimal values has shifted forward (compare the curve with 81 nodes in Fig. 4). This means that the more irregularity index, the larger sub-domain is needed. Such an effect can be taken into account by defining an irregularity factor which is to be multiplied to the optimal values given in Table 1. We have performed such a study for several patches with different irregularity indices and Poisson's ratios. The results are summarized in Table 2.

#### 4.4 The step by step GFPM procedure with optimal parameters

In this section we summarize the GFPM procedure for general 2D elasticity problems:



**Fig. 8** The effect of irregularity on patch behavior, **a** a sample of patch with  $S_{\max} = 0.175$  when 9 nodes are considered in each cloud, **b** a sample of patch with  $S_{\max} = 0.36$  when 9 nodes are considered in each

cloud, **c** variation of  $\lambda_{\max}/N_p$  versus  $\bar{a}$ , for a some sampling irregularity indices ( $\nu = 0.3$ )

- A grid of points/nodes is considered on the solution domain.
- Decision is made on the order of the polynomial to be used for the interpolation and accordingly the minimum number of nodes needed in the clouds.
- On each node of the domain a cloud of nodes with the minimum number of nodes is constructed. In this paper we have followed the simple cloud construction explained in Sect. 3.3.
- The procedure of WLS explained in Sect. 2, is followed to evaluate matrices  $\tilde{\mathbf{A}}^c$  and  $\tilde{\mathbf{B}}^c$  defined in relations (9) and (10).
- Based on the total number of nodes present in the cloud, a preliminary sub-domain size is chosen from Table 1 (possibly with interpolation/extrapolation between the values).
- The irregularity index is computed as (49).
- From the content of Table 2, an appropriate factor is chosen (use interpolation if needed). The sub-domain size chosen in the previous step is then multiplied by the evaluated factor to obtain the final optimal sub-domain size.
- With the sub-domain size in hand, the matrix  $\tilde{\mathbf{G}}^c$  defined in (23) is constructed for the evaluation of the corresponding rows of  $\mathbf{K}$  matrix in (38). The explicit form of the  $\tilde{\mathbf{G}}^c$  matrix is given in the “Appendix” (for clouds intersecting the boundaries with tractions, an equivalent formulation given in (32), using appropriate matrices in the “Appendix”, may be used).
- Having accomplished steps 3 to 8 for all nodes, the coefficient matrix  $\mathbf{K}$  is evaluated.
- The essential boundary conditions, as (35), are applied to the system of equations.
- The nodal displacements are evaluated by solving the matrix equation in (38).

The results of the above procedure are shown in the next section.

## 5 Numerical results

Here we present the results of the GFPM applied to some well-known benchmark problems. In the examples we shall present convergence graphs of errors defined as

$$e_u = \sqrt{\frac{\sum_{k=1}^{N_p} (\mathbf{u}_k - \mathbf{u}_k^{\text{exact}})^T (\mathbf{u}_k - \mathbf{u}_k^{\text{exact}})}{N_p}},$$

$$e_\sigma = \sqrt{\frac{\sum_{k=1}^{N_p} (\boldsymbol{\varepsilon}_k - \boldsymbol{\varepsilon}_k^{\text{exact}})^T (\boldsymbol{\sigma}_k - \boldsymbol{\sigma}_k^{\text{exact}})}{N_p}}, \quad (50)$$

$$\boldsymbol{\varepsilon} = \mathbf{S}\mathbf{u}, \quad \boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\varepsilon}$$

In above, the superscript “exact” has been used to denote the exact quantities.

*Example 1* The Timoshenko cantilever beam is revisited here. The exact solution can be found in [27]

$$u = \frac{P}{6EI} \left( y - \frac{h}{2} \right) \left[ (6Lx - 3x^2) + (2 + \nu) \left( y^2 - hy \right) - \frac{4 + 5\nu}{4} h^2 \right], \quad (51)$$

$$v = -\frac{P}{6EI} \left[ (3Lx^2 - x^3) + 3\nu (L - x) \left( y - \frac{h}{2} \right)^2 \right] \quad (52)$$

where  $u$  and  $v$  are displacement components along  $x$  and  $y$ ,  $\nu$  is Poisson’s ratio,  $E$  is Young’s modulus,  $h$  and  $L$  are height and length of the beam and  $I = \frac{h^3}{12}$ . In this example we use  $E = 1,000$ ,  $\nu = 0.3$ ,  $h = 1$ ,  $L = 5$  and  $P = 1$ . The end load is the resultant of a parabolic distribution of shear stresses. Figure 9a shows the beam configuration and the grid of nodes used. For the numerical solution all horizontal displacements are restrained at the fixed end and the prescribed values are obtained from the exact solution.

**Table 2** Irregularity factors for different Poisson's ratios

Poisson's ratio	$S = 0$	$S = 0.01$	$S = 0.1$	$S = 0.2$	$S = 0.3$	$S = 0.5$
$\nu = 0.05$	1	1.02	1.2	1.4	1.6	2
$\nu = 0.15$	1	1.12	1.3	1.5	1.7	2.1
$\nu = 0.30$	1	1.12	1.31	1.52	1.73	2.15
$\nu = 0.45$	1	1.02	1.21	1.42	1.63	2.05
Suggested for all $\nu$	1	1.1	1.3	1.5	1.7	2.1

As is seen, the exact solution is a third order polynomial and, therefore, the application of FPM and GFPM with third order polynomial for interpolation leads to reproduction of the exact solution (not shown here). We attempt to solve the problem with second order polynomials. The results are presented in Fig. 9b. In the figure we have included the results of a non-optimized sub-domain size, e.g.  $\bar{a} = 0.5$ . The figure clearly shows the superiority of GFPM to FPM especially when optimal sub-domain sizes have been employed.

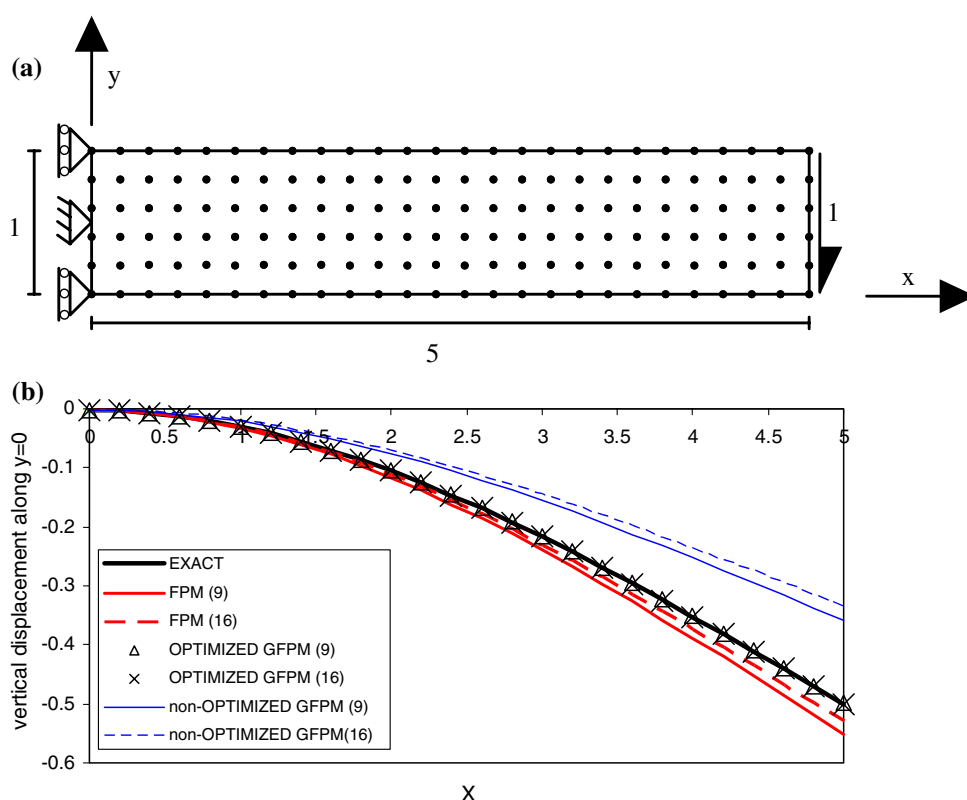
The results of the convergence study are given in Figs. 10 and 11. Figure 10a and b depict the results for error norms defined in (50) for displacements and strains/stresses when regular node distributions are used. It can be seen that the use of GFPM leads to better accuracy. In Fig. 11, however, it can be seen that both methods perform equally when irregular node distributions are employed for the solutions. The overall conclusion is that GFPM performs equally or better than FPM in this example.

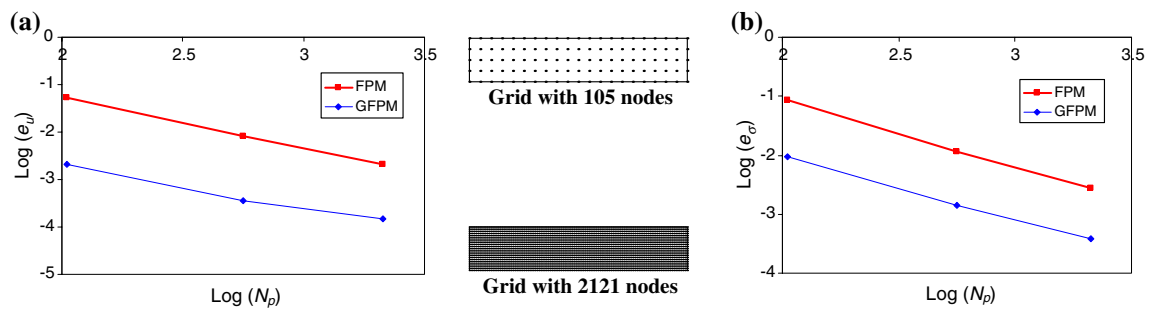
**Example 2** A plane stress elasticity problem on a square domain of  $2 \times 2$  is considered with following exact solution;

$$u_{ex} = \sinh(x) \sin(y), \quad v_{ex} = \cosh(x) \cos(y) \quad (53)$$

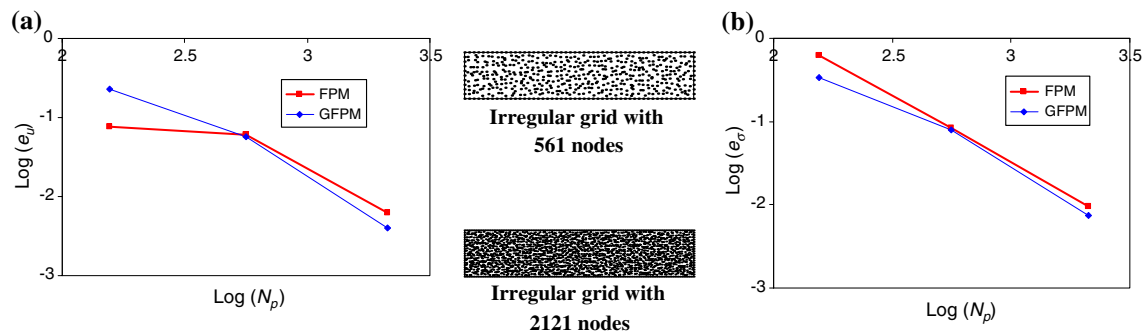
A set of regular grids with  $4 \times 4$ ,  $19 \times 19$ ,  $34 \times 34$  nodes and irregular grids with 16, 361, and 1,156 nodes are considered for the numerical solutions. The problem is solved under Dirichlet boundary conditions. Convergence plots of the norms defined in (50) are given in Fig. 12. In Fig. 12a it can be seen that both GFPM and FPM perform equally in regular grids when the norm of strains/stress is of concern. However, Fig. 12b shows that the accuracy of displacements in FPM is better than that of the GFPM although the convergence rates are similar. For the irregular grids the results of GFPM are superior to those of FPM as Fig. 12c and d depict. Here again the overall conclusion is the same as that in the first example.

**Fig. 9** The results obtained for Timoshenko beam problem. The numbers in the parentheses indicate the minimum number of nodes per cloud; **a** the grid of nodes used, **b** vertical displacement component  $v$  along  $y = 0$ . The results of non-optimized form of GFPM are taken for  $\bar{a} = 0.5$

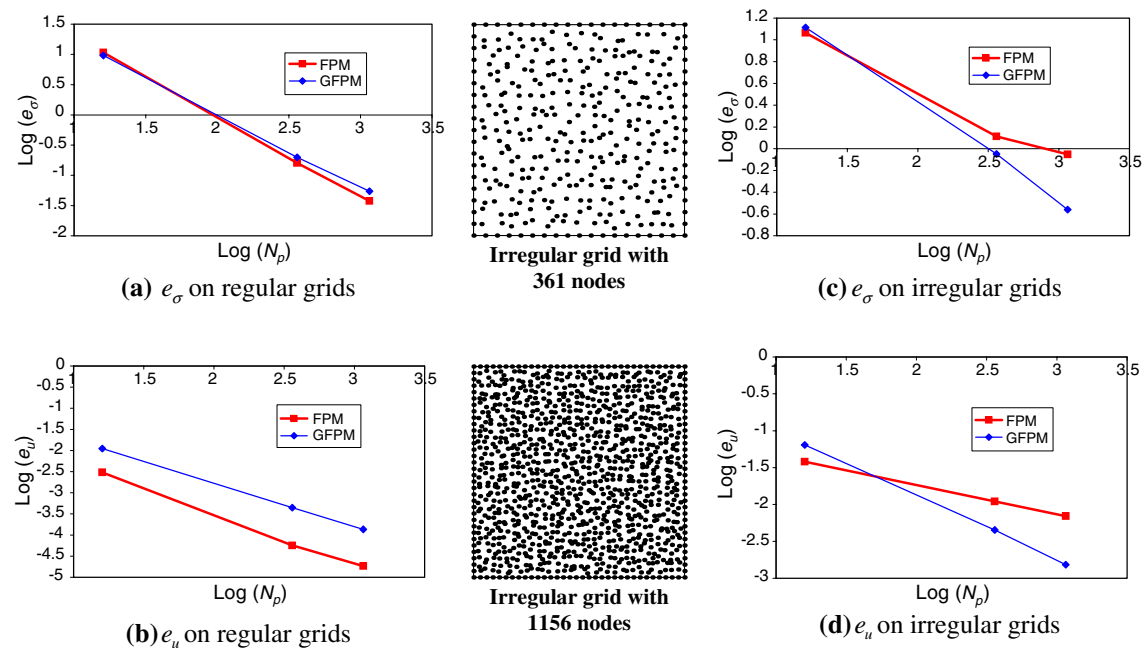




**Fig. 10** Convergence of the results for Timoshenko beam problem with regular grids; **a** convergence plot for error of displacements, **b** convergence plot for strains/stresses The minimum number of nodes in each cloud is 9

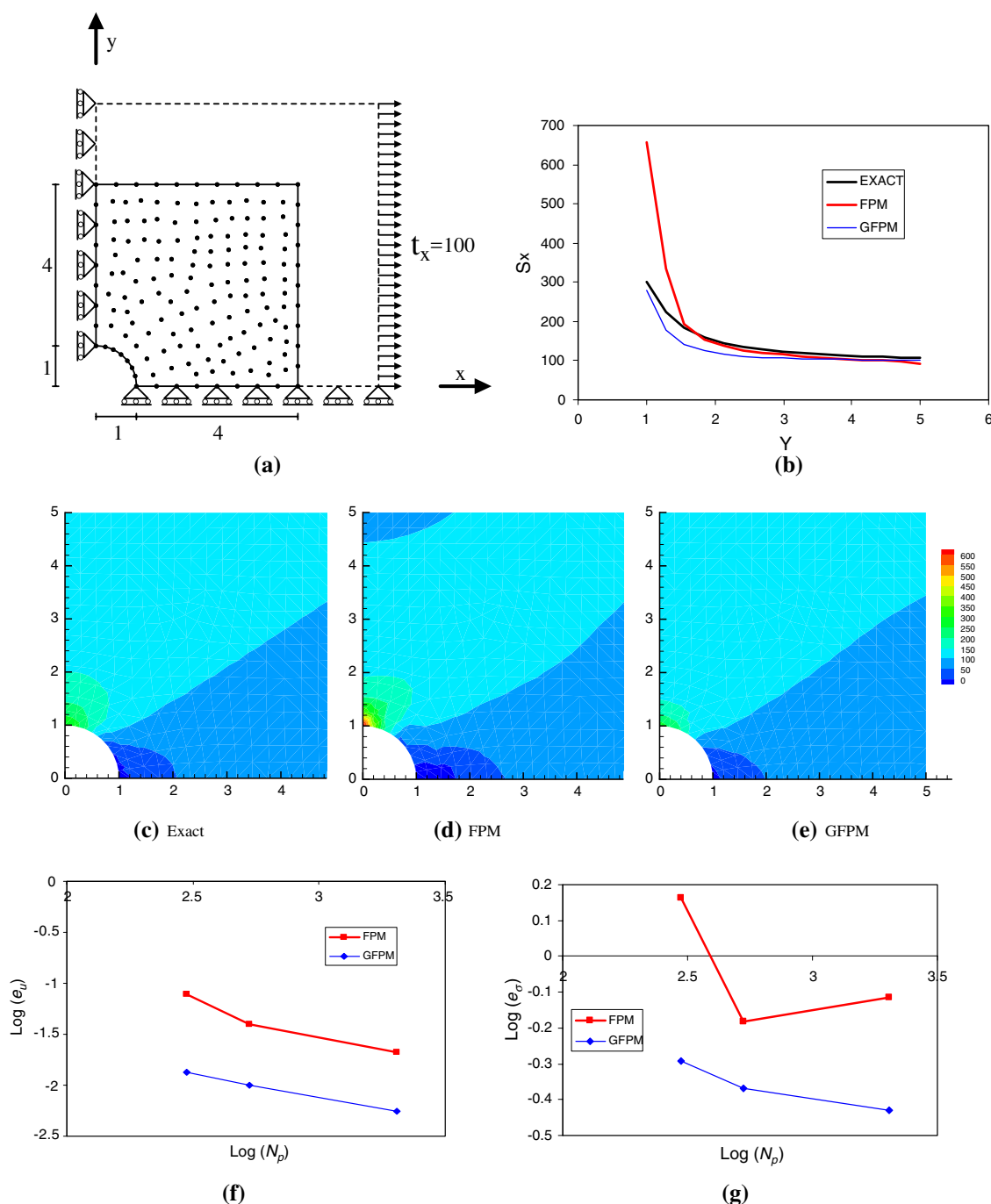


**Fig. 11** Convergence of the results for Timoshenko beam problem with irregular grids; **a** convergence plot for error of displacements, **b** convergence plot for strains/stresses The minimum number of nodes in each cloud is 9



**Fig. 12** Convergence of the results for Example 2; **a**, **b** convergence plots for error of displacements and strains obtained on regular grids; **c**, **d** convergence plots for error of displacements and strains obtained on irregular grids. The minimum number of nodes in each cloud is 9





**Fig. 13** The results obtained for the perforated plate; **a** the geometry and the grid of 298 nodes with 24 nodes/cloud, **b** variation of  $\sigma_x$  along y-axis, **c** contour plot of exact  $\sigma_x$ , **d** contour plot of  $\sigma_x$  obtained by FPM,

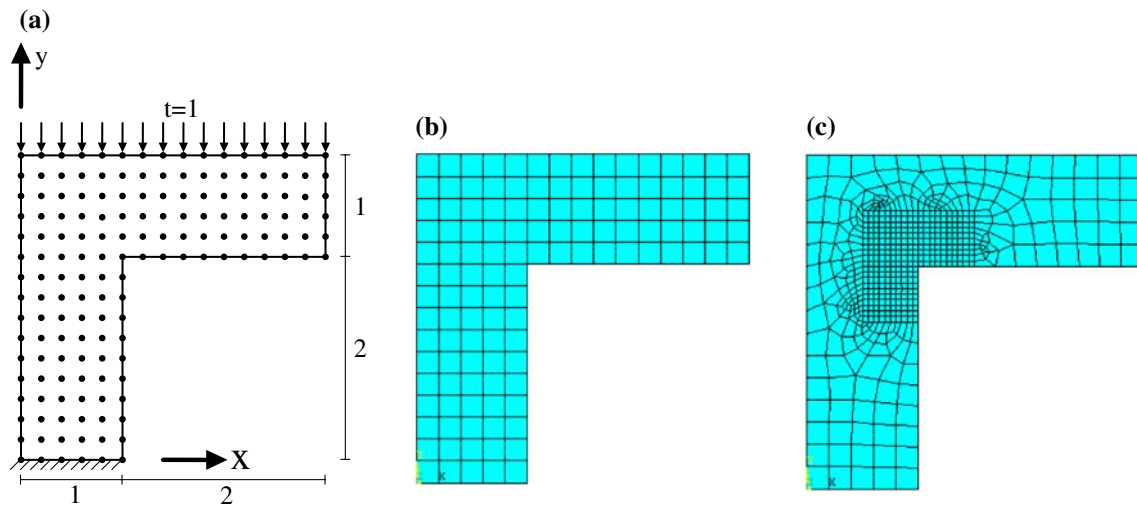
**e** contour plot of  $\sigma_x$  obtained by GPFM, **f** convergence plot for error of displacements, **g** convergence plot for strains/stresses

**Example 3** A perforated plate under tension is considered with plane stress condition. A quarter of the plate is solved using the tractions evaluated from the exact stress field (Fig. 13). The exact solution of the problem is given as

$$\sigma_x = t_x \left\{ 1 - \frac{r_0^2}{r^2} \left[ \frac{3}{2} \cos(2\theta) + \cos(4\theta) \right] + \frac{3r_0^4}{2r^4} \cos(4\theta) \right\} \quad (54)$$

$$\sigma_y = -t_x \left\{ \frac{r_0^2}{r^2} \left[ \frac{1}{2} \cos(2\theta) - \cos(4\theta) \right] + \frac{3r_0^4}{2r^4} \cos(4\theta) \right\} \quad (55)$$

$$\sigma_{xy} = -t_x \left\{ \frac{r_0^2}{r^2} \left[ \frac{1}{2} \sin(2\theta) + \sin(4\theta) \right] - \frac{3r_0^4}{2r^4} \sin(4\theta) \right\} \quad (56)$$



**Fig. 14** The geometry and the grids of nodes used for Example 4; **a** the grid of 156 node used for FPM and GFPM, **b** the mesh of linear quadrilaterals with 156 nodes (312 DOF) used for FE solution, **c** the mesh of quadratic quadrilaterals with 1,742 nodes (3484 DOF) used for FE solution

$$u_r = \frac{t_x}{4G} \left\{ r \left[ \frac{(\kappa - 1)}{2} + \cos(2\theta) \right] + \frac{r_0^2}{r} [1 + (1 + \kappa) \cos(2\theta)] - \frac{r_0^4}{r^3} \cos(2\theta) \right\} \quad (57)$$

$$u_\theta = \frac{t_x}{4G} \left[ (1 - \kappa) \frac{r_0^2}{r} - r - \frac{r_0^4}{r^3} \right] \sin(2\theta) \quad (58)$$

where

$$G = \frac{E}{2(1 + \nu)}, \quad \kappa = \frac{3 - \nu}{1 + \nu} \quad (59)$$

In above equations,  $r_0$  is the radius of the hole and  $(r, \theta)$  are the polar coordinates of the points with origin at the center of the hole. In this example we use  $E = 1,000$ ,  $\nu = 0.3$ ,  $t_x = 100$ . Figure 13b–g show some of the results obtained from the solution of the problem by the application of GFPM and FPM. In this example the minimum number of nodes per cloud is 24. Contour plot of  $\sigma_x$  is given in Fig. 13c–e for the three solutions (the grid used consists of 298 nodes). It can be seen that the accuracy of the results in the new method is significantly more than that of the original FPM. This effect can also be understood from the convergence plots for the errors given in Fig. 13f and g. This example also shows that the GFPM results are better than those of FPM.

**Example 4** The final example is of an L-shape domain with tractions at the top (see Fig. 14a). The bottom edge is restrained against movement in all directions. The material properties are the same as those in example 1. Since the exact solution is not known, here the problem has been solved with FEM, FPM and GFPM. Prior to analyze the results it is worthwhile to compare the CPU times used for the three methods. The codes are written in *Visual C++* environment.

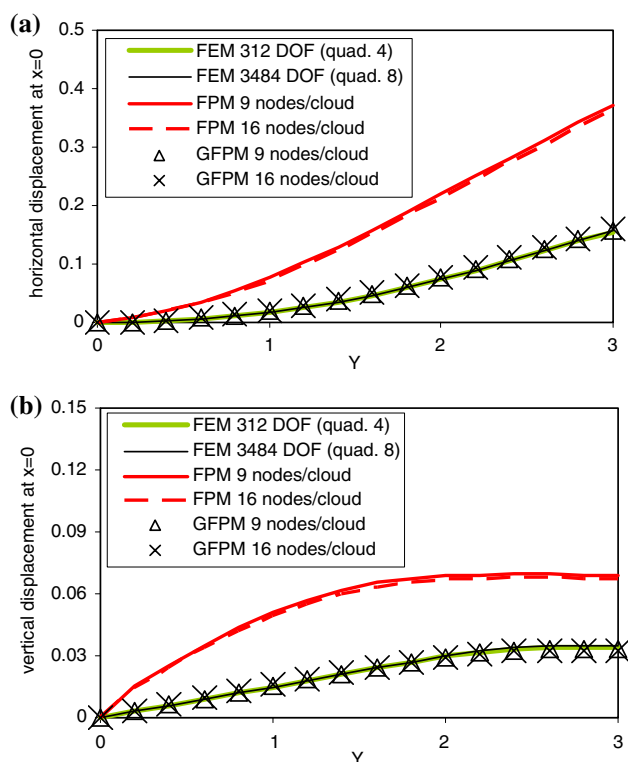
**Table 3** The CPU time used for solution of Example 4

Method	FEM (linear quads)	FPM (9 node/cloud)	GFPM (9 node/cloud)
CPU time (ms)	734	687	656

The processor is of *AMD Athlon (tm)* type equipped with a *64 Processor 3700+*. The computation times are calculated after reading the data for the geometry and the node coordinates (for grids with 156 nodes shown in Fig. 14a, b). The CPU times are listed in Table 3.

It can be seen that the order of CPU times are similar. It should also be mentioned that although in Fig. 14b the elements are of regular shape, to make a fair comparison we have solved the problem as a general FE one with an irregular mesh (i.e. for each element we have performed the integrations needed for the evaluation of the element stiffness). The reason lies in the fact that generally the CPU time for FPM and GFPM does not significantly differ for regular and irregular grids. We have also used minimum 9 nodes in each cloud in both FPM and GFPM so that the bandwidths of the coefficient matrices in the three methods become nearly equal (note that, regardless of the nodes at boundaries, in FE solution with square linear elements each of the hat shape-functions covers 9 nodes approximately). Table 3 shows that GFPM is as cheap as FPM.

The results of the numerical solutions are shown in Fig. 15a and b. Due to the presence of singularity at the reentrant corner, all the numerical solutions are prone to pollution error [28]. In order to alleviate the effect of the singularity in our reference solution, here the FE results, and also to improve the performance of the elements in bending,



**Fig. 15** The results obtained for the L shape domain; **a** variation of horizontal displacement along y-axis, **b** variation of vertical displacement along y-axis

we have also included the results obtained from a mesh of quadratic elements refined at the reentrant corner as shown in Fig. 15c. As is seen the results of GFPM are very close to those of FEM. Although it is not possible to measure the errors in this example, it seems that the results of FPM are more erroneous than those of FEM and GFPM. The reason for such a conclusion is that in FPM a set of equations, either in the form of equilibrium or traction conditions as in (17), must be satisfied at the reentrant corner where the exact gradients of displacements are not defined.

## 6 Conclusions

A new mesh-less method has been presented. The method is based on the equilibration of sub-domains, and in this respect may be regarded as the generalized form of the finite point method (GFPM). The formulation of the method is based on a weak form of a weighted residual approach with Heaviside step functions as the weights. The use of such weight functions leads to simple boundary integral equations which can be evaluated explicitly without the need for numerical integration when WLS is employed for interpolation. The order of differentiation of the shape-functions needed in GFPM is less than that in FPM and this may be regarded as one of the advantages of GFPM. Other advantages of GFPM are; its capability to model point loads and having a unique formu-

lation at corners where tractions are present at two faces of the boundary.

The sub-domains used in GFPM do not necessarily cover the whole solution domain. The optimal sizes of the sub-domains are determined through a generalized form of patch test. The final results for generic elasticity problems (with different Poisson's ratios) have been given in a series of tables. The content of the tables can be used for general problems as long as the cloud definition, including the weight and polynomial used, etc., are similar to the present work. For other cases, not included in this study, the step by step procedure of the patch test has been explained in details. Several sample problems have been solved to compare the performance of GFPM with those of FPM and FEM. One of the problems solved includes reentrant point which is the sources of singularity in the problems. The numerical experiments show that GFPM is capable of giving comparable results to those of FE solution. It has been found that GFPM performs equally or, in many cases, better than FPM while the costs of the computations in the two methods are similar.

## Appendix

In this appendix we present the closed forms of the integrals needed for equilibration of a sub-domain. Suppose that the sub-domain is of a polygon shape and the boundary integral (23) is to be evaluated by a summation of some integrals on lines as

$$\int_{\Gamma_i^s} \tilde{\mathbf{n}} \hat{\sigma}^c d\Gamma = \sum_m \int_0^1 \tilde{\mathbf{n}}_m \hat{\sigma}^c L_m d\xi \quad \hat{\sigma}^c = \mathbf{D} \mathbf{S} \hat{\mathbf{u}}^c \quad (\text{A-1})$$

where  $L_m$  is the length of the  $m$ th boundary segment and  $\xi$  is a local normalized coordinate. Also  $\tilde{\mathbf{n}}_m$  contains the components of unit normal to the boundary segment which is written as

$$\tilde{\mathbf{n}}_m = \begin{bmatrix} n_{mx} & 0 & n_{my} \\ 0 & n_{my} & n_{mx} \end{bmatrix} \quad n_{mx} = \frac{\Delta y_m}{L_m}, \quad n_{my} = -\frac{\Delta x_m}{L_m}, \quad L_m = \sqrt{(\Delta x_m)^2 + (\Delta y_m)^2} \quad (\text{A-2})$$

In above  $\Delta y_m = y_{m2} - y_{m1}$ ,  $\Delta x_m = x_{m2} - x_{m1}$  with  $(x_{mi}, y_{mi})$   $i = 1, 2$  being the coordinates of the two end points. Substitution of (A-2) in (A-1) leads to the following sets of equations for one segment

$$\int_0^1 \tilde{\mathbf{n}}_m \hat{\sigma}^c L_m d\xi = \int_0^1 \left\{ \begin{bmatrix} \hat{\sigma}_x^c \Delta y_m - \hat{\tau}_{xy}^c \Delta x_m \\ \hat{\tau}_{xy}^c \Delta x_m - \hat{\sigma}_y^c \Delta y_m \end{bmatrix} \right\} d\xi \quad (\text{A-3})$$

From the second relation in (A-1) one may write

$$\begin{Bmatrix} \hat{\sigma}_x^c \\ \hat{\sigma}_y^c \\ \hat{\tau}_{xy}^c \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} & 0 \\ D_{21} & D_{22} & 0 \\ 0 & 0 & D_{33} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \bar{\mathbf{P}}^c \bar{\mathbf{A}}^c \bar{\mathbf{B}}^c \bar{\mathbf{u}}^c \quad (\text{A-4})$$

Now supposing that the polynomial used is of second order, i.e.

$$\begin{aligned} \mathbf{P}(x) &= \{1 \ x \ y \ x^2 \ xy \ y^2\} \quad x = (1 - \xi)x_{m1} + \xi x_{m2}, \\ y &= (1 - \xi)y_{m1} + \xi y_{m2} \end{aligned} \quad (\text{A-5})$$

then the final integrations required are written as

$$\int_0^1 \bar{\mathbf{n}}_m \hat{\sigma}^c L_m d\xi = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{bmatrix} \bar{\mathbf{A}}^c \bar{\mathbf{B}}^c \bar{\mathbf{u}}^c \quad (\text{A-6})$$

where

$$\mathbf{C}_{11} = \left\{ \begin{aligned} &0 \quad \Delta y_m D_{11} \quad -\Delta x_m D_{33} \quad \Delta y_m D_{11} X_m \\ &\frac{1}{2}(\Delta y_m D_{11} Y_m - \Delta x_m D_{33} X_m) \quad -\Delta x_m D_{33} Y_m \end{aligned} \right\} \quad (\text{A-7})$$

$$\mathbf{C}_{12} = \left\{ \begin{aligned} &0 \quad -\Delta x_m D_{33} \quad \Delta y_m D_{12} \quad -\Delta x_m D_{33} X_m \\ &\frac{1}{2}(\Delta y_m D_{12} X_m - \Delta x_m D_{33} Y_m) \quad \Delta y_m D_{12} Y_m \end{aligned} \right\} \quad (\text{A-8})$$

$$\mathbf{C}_{21} = \left\{ \begin{aligned} &0 \quad -\Delta x_m D_{21} \quad \Delta y_m D_{33} \quad -\Delta x_m D_{21} X_m \\ &\frac{1}{2}(-\Delta x_m D_{21} Y_m + \Delta y_m D_{33} Y_m) \quad \Delta y_m D_{33} Y_m \end{aligned} \right\} \quad (\text{A-9})$$

$$\mathbf{C}_{22} = \left\{ \begin{aligned} &0 \quad \Delta y_m D_{33} \quad -\Delta x_m D_{22} \quad \Delta y_m D_{33} X_m \\ &\frac{1}{2}(-\Delta x_m D_{22} X_m + \Delta y_m D_{33} Y_m) \quad -\Delta x_m D_{22} Y_m \end{aligned} \right\} \quad (\text{A-10})$$

$$X_m = x_{m1} + x_{m2}, \quad Y_m = y_{m1} + y_{m2} \quad (\text{A-11})$$

For a rectangular domain the final result of the summation in (A-1) may be given as

$$\int_{\Gamma_i^s} \bar{\mathbf{n}} \mathbf{D} \hat{\mathbf{u}}^c d\Gamma = \sum_{m=1}^4 \int_0^1 \bar{\mathbf{n}}_m \hat{\sigma}^c L_m d\xi = \underbrace{\begin{bmatrix} \bar{\mathbf{G}}_{11}^c & \bar{\mathbf{G}}_{12}^c \\ \bar{\mathbf{G}}_{21}^c & \bar{\mathbf{G}}_{22}^c \end{bmatrix}}_{\bar{\mathbf{G}}^c} \bar{\mathbf{A}}^c \bar{\mathbf{B}}^c \bar{\mathbf{u}}^c \quad (\text{A-12})$$

in which

$$\bar{\mathbf{G}}_{11}^c = \{0 \ 0 \ 0 \ 2abD_{11} \ 0 \ 2abD_{33}\} \quad (\text{A-13})$$

$$\bar{\mathbf{G}}_{12}^c = \{0 \ 0 \ 0 \ 0 \ ab(D_{12} + D_{33}) \ 0\} \quad (\text{A-14})$$

$$\bar{\mathbf{G}}_{21}^c = \{0 \ 0 \ 0 \ 0 \ ab(D_{21} + D_{33}) \ 0\} \quad (\text{A-15})$$

$$\bar{\mathbf{G}}_{22}^c = \{0 \ 0 \ 0 \ 2abD_{33} \ 0 \ 2abD_{22}\} \quad (\text{A-16})$$

with  $a = \bar{a}L_1$  and  $b = \bar{b}L_2$  being the sizes of the rectangle along  $x$  and  $y$  axes, respectively.

## References

- Duarte CA (1995) A Review of some meshless methods to solve partial differential equations. TICAM Report 95-06, Texas Institute for Computational and Applied Mathematics
- Li S, Liu WK (2002) Meshfree particle methods and their applications. *Appl Mech Rev* 54:1–34
- Nguyen VP, Rabczuk T, Bordas S, Duflot M (2008) Meshless methods: a review and computer implementation aspects. *Math Comput Simul* (in press)
- Boroomand B, Tabatabaei AA, Oñate E (2005) Simple modifications for stabilization of the finite point method. *Int J Numer Methods Eng* 63:351–379
- Perrone N, Kao R (1975) A general finite difference method for arbitrary meshes. *Comput Struct* 5:45–58
- Nayroles GB, Touzot, Villon P (1992) Generalizing the finite element method: diffuse approximation and diffuse elements. *Comput Mech* 10:307–318
- Belytschko T, Lu YY, Gu L (1994) Element-free Galerkin methods. *Int J Numer Methods Eng* 37:229–256
- Liu WK, Jun S, Li S, Adee J, Belytschko T (1995) Reproducing kernel particle methods for structural dynamics. *Int J Numer Methods Eng* 38:1655–1679
- Fasshauer GE (1997) Solving partial differential equations by collocation with radial basis functions. In: Mehaute AL, Rabut C, Schumaker LL (eds) *Surface fitting and multiresolution methods*, pp 131–138
- Oñate E, Idelsohn S, Zienkiewicz OC, Taylor RL (1996) A finite point method in computational mechanics—applications to convective transport and fluid flow. *Int J Numer Methods Eng* 139:3839–3866
- Atluri SN, Zhu T (2000) The meshless local Petrov–Galerkin (MLPG) approach for solving problems in elasto-statics. *Comput Mech* 25:169–179
- Aluru NR (2000) A point collocation method based on reproducing kernel approximations. *Int J Numer Methods Eng* 47:1083–1121
- Huerta A, Vidal Y, Villon P (2004) Pseudo-divergence-free element free Galerkin method for incompressible fluid flow. *Comput Methods Appl Mech Eng* 193:1119–1136
- Krongauz Y, Belytschko T (1998) EFG approximation with discontinuous derivatives. *Int J Numer Methods Eng* 41:1215–1233
- Carpinteri A, Ferro G, Ventura G (2002) The partition of unity quadrature in meshless methods. *Int J Numer Methods Eng* 54: 987–1006
- Duarte CA, Oden JT (1996) HP clouds—an HP meshless method. *Int J Numer Methods Eng* 12:673–705
- Kuchеров L, Tadmor EB, Miller RE (2000) Umbrella spherical integration: a stable meshless method for nonlinear solids. *Int J Numer Methods Eng* 69:2807–2847

18. Oñate E, Idelsohn S (1998) A mesh-free finite point method for advective-diffusive transport and fluid flow problems. *Comput Mech* 21(4/5):283–292
19. Oñate E, Sacco C, Idelsohn S (2000) A finite point method for incompressible flow problems. *Comput Vis Sci* 2:67–75
20. Löhner R, Sacco C, Oñate E (2002) A finite point method for compressible flow. *Int J Numer Methods Eng* 53:1765–1779
21. Kim DW, Kim Y-S (2003) Point collocation methods using the fast moving least square reproducing kernel approximation. *Int J Numer Methods Eng* 56:1445–1464
22. Lee S-H, Yoon Y-C (2004) Meshfree point collocation method for elasticity and crack problem. *Int J Numer Methods Eng* 61(1): 22–48
23. Sadeghirad A, Mohammadi S (2007) Equilibrium on line method (ELM) for imposition of Neumann boundary conditions in the finite point method (FPM). *Int J Numer Methods Eng* 69:60–86
24. Ortega E, Oñate E, Idelsohn S (2007) An improved finite point method for tridimensional potential flows. *Comput Mech* 40: 949–963
25. Boroomand B, Mossaiby F (2005) Generalization of robustness test procedure for error estimators. Part I: formulation for patches near kinked boundaries. *Int J Numer Methods Eng* 64:427–460
26. Boroomand B, Mossaiby F (2005) Generalization of robustness test procedure for error estimators. Part II: test results for error estimators using SPR and REP. *Int J Numer Methods Eng* 64: 461–502
27. Timoshenko SP, Goodier JN (1970) *Theory of elasticity*, 3rd edn. McGraw-Hill, New York
28. Babuška I, Strouboulis T, GangarajSK, Upadhyay CS (1997) Pollution error in the h-version of the finite element method and the local quality of the recovered derivatives. *Comput Methods Appl Mech Eng* 140:1–39